Issues in Information Hiding Transform Techniques*

1 INTRODUCTION

Information hiding has emerged as an exciting and important research field. Information hiding not only complements the traditional obfuscation techniques, (e.g., [17]) but also brings to it new prospects. By its definition, information hiding hides a message (the *embedded* message) under a *cover* message to yield the *stego*-message. Much of the research in information hiding has focused upon steganography and watermarking. Steganography refers to methods that are used to transmit the embedded message without an observer being aware that there is an embedded message in the cover message. The embedded message may be fragile - it is easily broken in the face of attacks. With respect to steganography, *robustness* is not a critical property. *Transparency* is! The similar field of watermarking is to embed a "watermark" for the purpose of authentication, a crucial step for copyright protection and tamper proofing. The embedded watermark may not be transparent in the sense that it is perceivable, but it must not be easily removed from the stego message. The embedded watermark is usually required to be semi-fragile (i.e., destroyed if changes exceed a limit) or robust. Johnson et. al. [8] nicely state (their concern is images) that "Traditional steganography conceals information; watermarks extend information and may be considered attributes of the cover image."

In our present experiments, digital images are used as the cover message in which we embed the hidden information. Two common modes of embedding are spatial embedding and transform embedding. **Spatial embedding** inserts messages into image pixels, usually in the least significant bits[1] (LSB)[2] [10]. LSB embedding has the merit of simplicity, but suffers from the lack of robustness. LSB embedding is susceptible to image-processing type of attacks. Error-correction coding has been proposed for enhancing the robustness [9][13], but its effectiveness is limited to low levels of noise. If spatial embedding involves higher order bits, one runs the very real risk of the steganography being

*Research supported by the Office of Naval Research.

[1]Early experiments of embedding messages under the least significant bits in audio steganography were performed by Kang [9].

[2]Abbreviations may be singular or plural.

detected, and for watermarking the concern is that the cover image might be degraded and/or the watermark may be easy to remove. In order to achieve robust hiding, researchers have invoked transform domain techniques (e.g., frequency space) [5]. **Transform embedding** embeds a message by modifying (selected) transform (e.g., frequency) coefficients of the cover message. Ideally, transform embedding has the effect in the spatial domain of apportioning the hidden information through different order bits in a manner that is robust, but yet hard to detect. Of course one must then be concerned with the detectability in the frequency domain, but at least the human visual system (HVS) may be fooled. Therefore, hiding in the frequency domain presents its own challenges (e.g., [5][7]). Since an attack, such as image processing, usually affects a certain band of transform coefficients, the remaining coefficients would remain largely intact. Hence, transform embedding is in general more robust than spatial embedding.

Extraction of the embedded message is often carried out by comparing the stego-message with the cover message. This is practical for watermarking, but one may not have the original cover message when dealing with a stego-message. Without a cover image, embedding may involve a stego-key. The stego-key would serve a similar purpose as the cover image in that it (hopefully) enables us to determine the hidden message. Also note that for message authentication, it may be sufficient only to prove the existence of the embedded message perhaps via a similarity measure (e.g., [5]). Also, In the absence of the original image, statistical methods based on detection probability have been proposed for extraction (e.g., [20]).

2 REVIEW

In this section, we will briefly review the three most commonly used transform techniques: DFT, DCT and Wavelet.

2.1 Discrete Fourier Transform: DFT

The DFT has its root in the Fourier series analysis. Recall that a time domain periodic function $f(t)$ can be decomposed into a series of sine (or cosine) wave functions, where each has frequency that is a multiple of a constant (i.e., the 1st harmonic ω_0).[3] The goal is finding the coefficient for each wave function. For the purpose of frequency domain analysis, the exponential Fourier series is used in places for sine or cosine series and its coefficient of the nth harmonic (i.e., $n\omega_0$) is given by $F_n = (1/P)\int_0^P f(t)\exp^{-\iota n\omega_0 t}\,dt$, where ι denotes the complex number $\sqrt{-1}$, P denotes the duration of a period and ω_0 is $2\pi/P$.

Consider the one-dimension discrete case in which N samples $f(0), f(T), \cdots, f(NT)$ are taken at the sampling rate T. The sampled sequence may not have a period, but in the DFT it is assumed that these N samples constitute a

[3]The constant ω_0 is needed to assure the orthogonality between two wave functions.

period. As a result, the period of the sampled sequence becomes NT and correspondingly, the constant frequency ω_0 is $2\pi/NT$. The discrete Fourier transform is obtained by substituting respectively ω_0 with $2\pi/NT$, t with kT, dt with T, P with NT and n with u in the exponential Fourier series, i.e.,

$$F(u) \;=\; \frac{1}{N}\sum_{k=0}^{N-1} f(kT)\exp^{-\iota 2\pi(\frac{uk}{N})} \qquad 0 \le u < N$$

where u is the index in the frequency domain. Here, the total number of frequency components is also N. The lowest frequency component of the DFT occurs at $u = 0$ and is 0. The highest frequency component can be determined from the Nyquist sampling theorem and its value is $\frac{1}{2T}$ Hz (or cycles/second). The index u which corresponds to the highest frequency component is $\frac{N}{2}$, right at the middle of the N frequency indices.[4] For the digital pictorial domain, the sampling interval T is measured in terms of, not time, but pixels between consecutive samplings. In the case of one pixel per sampling, i.e., $T = 1$, the highest frequency component becomes $\frac{1}{2}$ cycles/pixel.[5] The highest frequency (or, the bandwidth) has been used in computing the lower bound of the hiding capacity of a stego image, where the lower bound is computed from the Shannon's capacity measure of an additive white Gaussian noise (AWGN) channel[6] with the embedded message being viewed as the signal and the cover message as the noise. (e.g., [18][13]).

Let $I(i, j)$ denote the brightness value of the pixel at position (i, j) of an image. The 2D DFT is a natural extension of the 1D DFT by applying the 1D DFT to a 2D matrix twice, and its period is given by the dimension of the input image (i.e., $N \times M$), i.e.,

$$F(u,v) \;=\; \frac{1}{NM}\sum_{k=0}^{N-1}\sum_{l=0}^{M-1} I(k,l)\exp^{-\iota 2\pi(\frac{uk}{N}+\frac{vl}{M})} \;, 0 \le u < N;\; 0 \le v < M \;. \quad (1)$$

Its backward transform[7] is given by

[4] Recall that the effect of sampling at time interval T in time domain yields a series of replicas of the frequency spectral separated at $(2\pi)/T$ apiece in the frequency domain. The Nyquist sampling theorem says the maximum sampling interval T is reciprocally lower-bounded by the frequency bandwidth W, i.e., $(2\pi)/T \ge 2W$. Let u_{max} denote the highest frequency index. We have $(2\pi)/T = 2(u_{max}\omega_0)$ or $2(u_{max})(2\pi/NT)$. Thus, u_{max} is equal to $\frac{N}{2}$. The highest frequency (i.e., $u_{max}\omega_0$) is $\frac{\pi}{T}$ radians/second or $\frac{1}{2T}$ cycles/second.

[5] For a digital image, the highest frequency of one direction may differ from that of the other direction. Here, the two highest frequency components are assumed to be the same.

[6]
$$C = W log_2(1 + \frac{S}{N}),$$
where W is the bandwidth, S denotes the energy measure of the signal, and N denotes the energy measure of the noise.

[7] More precisely, the backward transform should be the inverse mapping F^{-1}. We use $I(k, l)$ instead of $F^{-1}(k, l)$ for convenience.

$$I(k,l) \;=\; \sum_{u=0}^{N-1}\sum_{v=0}^{M-1} F(u,v)\exp^{\iota 2\pi(\frac{uk}{N}+\frac{vl}{M})} \;\; , 0 \le k < N;\; 0 \le l < M \qquad (2)$$

EQ(1) can also be written in the matrix form,

$$\begin{pmatrix} V_1 \\ \cdots \\ V_N \end{pmatrix} (I)_{MN} \begin{pmatrix} U_1^T & \cdots & U_M^T \end{pmatrix} \qquad (3)$$

where V_i and U_j^T denote $\{\exp^{-\iota 2\pi(\frac{v_i 0}{M})}, \exp^{-\iota 2\pi(\frac{v_i 1}{M})}, \cdots, \exp^{-\iota 2\pi(\frac{v_i (M-1)}{M})}\}$ and $\{\exp^{-\iota 2\pi(\frac{u_j 0}{N})}, \exp^{-\iota 2\pi(\frac{u_j 1}{N})}, \cdots, \exp^{-\iota 2\pi(\frac{u_j (N-1)}{N})}\}$, respectively. Note that the DFT obeys the property of symmetry i.e, $F(u,v) = F^*(N-u, N-v)$[8], which can be seen by replacing u and v with $N-u$ and $N-v$ in $\exp^{-\iota 2\pi(\frac{uk}{N}+\frac{vl}{M})}$. The property of symmetry is useful for plotting the result of the DFT as shown in the next section. The 2D DFT is a common instrument for analyzing hiding capacity and is presently available in our xv tool.

2.2 Discrete Cosine Transform: DCT

The DCT had been the major mathematical framework for image compression in JPEG until JPEG2000 was introduced. The DCT improves the DFT by eliminating the high frequency components induced by the sharp discontinuities at the boundary between two consecutive periods in the time (or spatial) domain of a periodic signal. To represent the sharp value change, it needs non-zero high frequency DFT coefficients. If for compression reasons all high frequency components of DFT, including those generated from the sharp discontinuities, are deleted, such deletion will cause distortion to the original image. To eliminate those undesirable high frequency components, the DCT concatenates a period with the mirrored image of its an adjacent period. This new period has twice the sample points, but no sharp value change at the boundary with its neighbors. Concatenation of one period and the mirror image of adjacent period defines an even function and hence, results in yielding an all real-valued transform code.[9] This is a big advantage in computation! The DCT can be obtained from the DFT of a mirrored 2N sample sequence, where the DCT is the first N sample points. The commonly used form of the DCT was derived from a class of discrete Chebyshev polynomials [1]. The derivation of the 2D

[8] $F^*(.,.)$ is the complex conjugate of $F(.,.)$

[9] Suppose a function, $g(t)$, whose domain is interval [0,P), is concatenated with its shifted mirror image, $g(2P-t)$. The Fourier transform of this concatenated function is given by $(1/2P)\int_0^{2P}(g(t)+g(2P-t))\exp^{-\iota n\omega_0 t}\,dt$, where $\exp^{(-)\iota n\omega_0 t} = cos(n\omega_0 t)+(-)sin(n\omega_0 t)$. It can be rewritten as $(1/2P)\left(\int_0^P g(t)\exp^{-\iota n\omega_0 t}\,dt + \int_P^{2P} g(2P-t)\exp^{-\iota n\omega_0 t}\,dt\right)$. By replacing $2P-t$ with t in the second term, the Fourier transform becomes $(1/P)\int_0^P g(t)cos(n\omega_0 t)dt$.

DCT code is similar to that of the DFT. The DCT code of an image brightness matrix $I(i,j)$ $(0 \leq i < N,\ 0 \leq j < M)$ is given by

$$S(u,v) \ = \ c(u,v) \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} I(i,j) cos \frac{\pi(2i+1)u}{2N} cos \frac{\pi(2j+1)v}{2M} \ , \qquad (4)$$

where $0 \leq u < N$ and $0 \leq v < M$, and c(u,v) is given by c(0,0)=$\sqrt{1/N}\sqrt{1/M}$, c(u,0)=$\sqrt{2/N}\sqrt{1/M}$, c(0,v)=$\sqrt{1/N}\sqrt{2/M}$, and c(u,v)=$\sqrt{2/N}\sqrt{2/M}$, $u, v > 0$. For each u and v, different values of $cos\frac{\pi(2i+1)u}{2N}$ $cos\frac{\pi(2j+1)v}{2M}$, $0 \leq i < N$ and $0 \leq j < M$, form a NxM DCT basis matrix. The DCT basis matrices are orthonormal. Coefficients produced from these base matrices are uncorrelated and hence can be processed independently. The backward DCT is shown below.

$$I(i,j) \ = \ \sum_{u=0}^{N-1} \sum_{v=0}^{M-1} c(u,v) S(u,v) cos \frac{\pi(2i+1)u}{2N} cos \frac{\pi(2j+1)v}{2M} \ . \qquad (5)$$

In JPEG, the DCT is applied to each block of 8X8 pixels from the input image, with the image being partitioned into a number of blocks [15].

2.3 Discrete Wavelet Transform: DWT

The wavelet transform (WT) has been adopted as the standard tool in JPEG 2000 still image compression as it produces a higher compression ratio than the DCT does [4]. Studies of image compression also show that the wavelet transform provides better frequency and time (spatial) resolution than other transform techniques do.

The DFT gives an excellent description of the frequency responses of a signal, but no information about when (where) particular frequency components occur in time (space). The Short-time Fourier Transform (STFT) improves the DFT by breaking the signal into intervals of fixed length and applying the Fourier analysis to each interval. A particular frequency response that occurs only at a certain interval can be captured with STFT. However, fixed length intervals have their restrictions. Although a short fixed length interval is good for identifying local variation in time (space), it is inadequate to describe frequency responses whose cycles exceed the length of the interval. The major changes from STFT to WT are perhaps the selection of base functions (e.g., the sinusoidal functions in Fourier transform) and the windowing operation. A base function of wavelet transform can be any function with zero mean and finite energy (called the *wavelet*).[10] The entire set of base functions are mutually orthonormal (like sinusoidal bases) and generated from a single base function (called the mother wavelet) by scaling and translation. In WT, a base function is locally applied

[10] That is, $\int \Psi(t)^2 dt \leq$ inf and hence, a base function is in vector space L_2. Because of the finite energy requirement, $\Psi(t)$ is restricted to a narrow band, which gives the wavelet its frequency localization capability [16]. A sine (cosine) function cannot be a base.

to a particular area of the signal at a time. Localization is realized through windowing, where the size of the window, indicating resolution, unlike the fixed interval used in STFT, is not a constant. Only the base function whose scale (or cycle) is compatible with the size of the window used. As a result, base functions of slower cycles are used under a larger window, while base functions of faster cycles are used under a shorter window.

In the case of data compression, the implementation of the DWT is similar to that of subband coding[16], where at each stage a coarse overall shape and details of the data obtained from the previous stage are derived. Encoding in the DWT proceeds with decomposition and downsampling. Decomposition separates data into frequency bands via high-pass and low-pass filtering. The functions of a high-pass filter are the WT base functions, while the functions of the low-pass filter are the complements of the base functions. Downsampling removes data which is not needed for future reconstruction. Decoding on the other hand involves up-sampling to adjust dimensionality and recombining data from different bands.

Call the output from high-pass and low-pass filtering the filtered transform coefficients. Let h, l and $\otimes$ denote the high-pass, low-pass and the convolution operation, respectively. Consider the case where the low-pass filter is a 2-tap averaging operator (i.e, l(0)=1/2, l(1)=1/2) and the high-pass filter is a difference operator (i.e., h(0)=1/2, h(1)=-1/2 - the Haar transform). Let $X = \{x_1, \cdots, x_8\}^T$. The outcomes of filtering are the high-filtered coefficients $h \otimes X$ and the low-filtered coefficients $l \otimes X$, i.e.,

$$l \otimes X = \frac{1}{2}\left[x_7 + x_0,\ x_0 + x_1, \cdots, x_5 + x_6,\ x_6 + x_7\right]^T \tag{6}$$

$$h \otimes X = \frac{1}{2}\left[x_0 - x_7,\ x_1 - x_0, \cdots, x_6 - x_5,\ x_7 - x_6\right]^T \tag{7}$$

The original signal can be reconstructed from those high-filtered and low-filtered coefficients by, for instance, adding them one by one and dividing the result of addition by 2. In fact, it can be shown that reconstruction needs just half the number of coefficients from each set and hence, each of the two sets is down-sampled to a half. If downsampling D is picking up every other coefficient from $l \otimes X$ and $h \otimes X$, it has the form

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{8}$$

The relationship between original data and the transform code is described in the matrix form as follows,

$$W_a\left[X\right] \ = \ \begin{bmatrix} DX_l \\ DX_h \end{bmatrix}\left[X\right] \tag{9}$$

where W_a is the DWT.

The wavelet transform may be applied to each set of filtered transform coefficients to obtain more detailed and coarser description. For instance, after downsampling, we have
Stage 1:

$$coarse: \quad \frac{1}{2}\left[x_7 + x_0,\ x_1 + x_2,\ x_3 + x_4,\ x_5 + x_6\right]^T \tag{10}$$

$$detail: \quad \frac{1}{2}\left[x_0 - x_7,\ x_1 - x_2,\ x_3 - x_4,\ x_5 - x_6\right]^T \tag{11}$$

We may continue the process recursively to get further decomposition.
Stage 2:

$$coarse: \quad \frac{1}{4}\left[x_7 + x_0 + x_1 + x_2,\ x_3 + x_4 + x_5 + x_6\right]^T \tag{12}$$

$$detail: \quad \frac{1}{4}\left[x_7 + x_0 - x_1 - x_2,\ x_3 + x_4 - x_5 - x_6\right]^T \tag{13}$$

Stage 3:

$$coarse: \quad \frac{1}{8}\left[x_7 + x_0 + x_1 + x_2 + x_3 + x_4 + x_5 + x_6\right]^T \tag{14}$$

$$detail: \quad \frac{1}{8}\left[x_7 + x_0 + x_1 + x_2 - x_3 - x_4 - x_5 - x_6\right]^T \tag{15}$$

The coefficient matrix is

$$\begin{aligned}[\frac{1}{8}(x_7+x_0+x_1+x_2+x_3+x_4+x_5+x_6),\ \frac{1}{8}(x_7+x_0+x_1+x_2-x_3-x_4-x_5-x_6),\\ \frac{1}{4}(x_7+x_0-x_1-x_2),\ \frac{1}{4}(x_3+x_4-x_5-x_6),\\ \frac{1}{2}(x_0-x_7),\ \frac{1}{2}(x_1-x_2),\ \frac{1}{2}(x_3-x_4),\ \frac{1}{2}(x_5-x_6)]^T\end{aligned}$$

Note that the first element of the coefficient matrix is the average of all values. For the 2D DWT (i.e., $W_a X W_a^T$), the transform codes of an image are divided into four pieces, often labeled as {LL, HL, LH, HH}. LL corresponds to the coefficients resulting from twice low-pass filtering and carries the most important information from the original image. Its size is just one quarter of the image. The remaining three pieces are the detailed components. Similar to the example shown above, for better compression result, the high and low filters are applied to the four (usually, just the LL) pieces.

3 DISCUSSION

In this section, we show our experimental results with transform embedding, and discuss two cases related to robustness and detection of embedded messages. Embedding is based on the following steps: (1) Apply the transform algorithm to the cover and the embedded data, (2) select the embedding method to combine the two sets of coefficients, and (3) apply the inverse transform to the combined coefficients to produce the stego image. In watermarking, extraction of the embedded message usually involves the subtraction of the coefficients of the cover from the coefficients of the stego, whereas in steganography, extraction may involve the use of the pre-assigned stego key.

3.1 Experimental Results

To illustrate transform domain hiding, we embed an image (Waterdrop) under a cover image (Washington Monument), where the two images are of the same size. Let F_e and F_c denote the transform code of the embedded and the cover images, respectively. (Note that, the embedded messages may not be transformed.) The embedding formula is in general described as

$$F_s(u,v) = F_c(u,v) + J(u,v) * F_e(u,v); \quad 0 \leq u \leq M, 0 \leq v \leq N$$

where $J(u,v)$ denotes the perceptual factor calculated for each frequency component [19]. In its simplistic form the $J(u,v)$ can be either additive (e.g., $J(u,v) = \alpha$), where α is an attenuation factor for adjusting the magnitude of embedded coefficients and $F_s = \alpha * F_e + F_c$, or multiplicative (e.g., $\alpha * F_c(u,v)$), where the coefficient of the cover, $F_c(u,v)$, is involved, and $F_s = F_c * (1 + \alpha * F_e)$. The advantage of embedding in the additive form is its efficient invertibility [5] for extraction. Not all coefficients of the cover are used for embedding. Transform coefficients of low frequency components that contain the most important overall information of the original image usually are excluded from being used for embedding. For instance, in [2], coefficients from the middle frequency (DWT) bands are randomly selected for embedding. In our current experiments, we set $J(u,v)$ to 1 and linearly combined the two sets of coefficients, i.e.,

$$F_s = \alpha * F_e + (1 - \alpha) * F_c,$$

in order to ensure that pixel values obtained from the inverse transformation will be in the proper dynamic range. (The scaling factor is chosen to be $\alpha = 0.05$.) Since addition in the Fourier domain results in addition in the time (spatial) domain, linear combination assures that image values extracted from F_s will not fall outside the allowed range. (On the other hand, linear combination does not make the most use of the transform domain, since embedding in one is basically equivalent to embedding in another.)

The results of our experiments are shown in Figure 1 to Figure 6. Comparing the original (Figure 1) and the stego (Figure 5), perceptually the two show no difference. The companion figures to the images are their corresponding DFT

matrices. Note that the coefficient at the left corner of a DFT matrix obtained from (1) should be the lowest frequency component (i.e., $u = 0,\ v = 0$ or the DC). However, because of the symmetric property of the DFT, it is customary to display the DC component at the center, and the further away from the center a DFT component is, the higher is its corresponding frequency. In our present display, the frequency component at (u, v) is moved to a new position by

$$\begin{array}{ll}((M/2-1)-u,\ (N/2-1)-v) & if\ 0 \le u < (M/2); 0 \le v < (N/2) \\ ((3M/2-1)-u,\ (N/2-1)-v) & if\ (M/2) \le u < M; 0 \le v < (N/2) \\ ((M/2-1)-u,\ (3N/2-1)-v) & if\ 0 \le u < M; (N/2) \le v < N \\ ((3M/2-1)-u,\ (3N/2-1)-v) & if\ (M/2) \le u < M; (N/2) \le v < N \end{array}$$

To further enhance the DFT display, a logarithmic transform is applied to adjust the dynamic range of coefficients and the result is normalized to be within the level of 0 to 255 (in order for our xv tool to display). Since the magnitude of the DC component is far larger than that of any other frequency component, the DC component is actually removed from the DFT image (seen as a black dot at the center).

*Fig*1. *the cover*

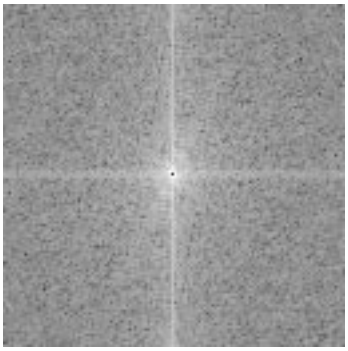

*Fig*2. *DFT of the cover*

*Fig*3. *the embedded*

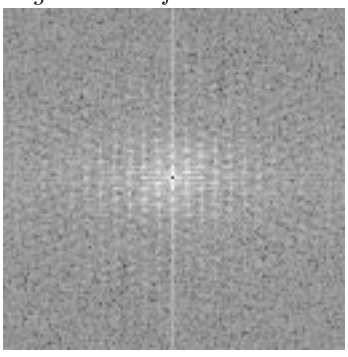

*Fig*4. *DFT of the embedded*

*Fig*5. *the stego*

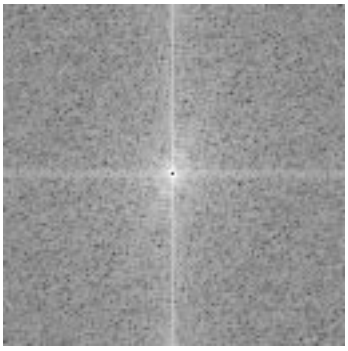

*Fig*6. *DFT of the stego*

At present, we have not yet implemented adaptive selection of transform coefficients. We do not suggest embedding spatial data (i.e., pixels) of the embedded image under the frequency coefficients of the cover (i.e., $I_e + T_c$) due to the fact that the frequency coefficients usually have a much larger dynamic range. Hence, changes to the frequency components (due to rounding and inverse transformation) can cause irremediable distortion to the embedded spatial data.

Extraction is implemented by reversing the embedding steps, i.e., $(F_s^{'} - (1 - \alpha)F_c)/\alpha = F_e^{'}$, where $'$ indicates the change of values due to image processing attacks. The embedded image extracted from the stego (Figure 7) also appears to be nearly identical to the original Waterdrop image. However, the significant reduction in magnitude of frequency coefficients during embedding taxes the quality when image compression is in order. On the right-handed side of Figure 7 is another extracted image (Figure 8) obtained from applying JPEG to the stego image. The grossly smeared image shows the need of more robust embedding.

*Fig*7. *the extracted embedded image*

*Fig*8. *JPEG* (*Quality* 75%)[11]

For comparison, Figures 9 & 10 show the extracted images in case the least 2 significant bits from the spatial domain are used for embedding [10].

[11]The quality value is expressed on the 0..100 scale recommended by Independent JPEG Group. It is related to the DCT quantization.

*Fig*9. *L2SB embedding* (*Quality* 100%)

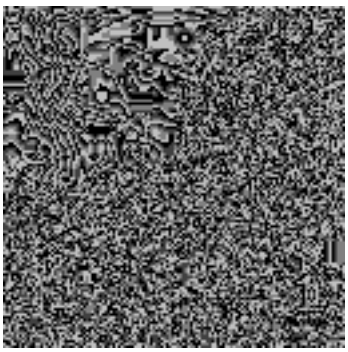

*Fig*10. *L2SB* (*Quality* 75%)

The outcome supports our observation that LSB embedding is susceptible to image processing attacks.

The result of embedding with the DWT is similar to that of the DFT and is shown in Figures 11&12. The DWT does *not* provide better robustness; robustness is not a property of transform algorithms.

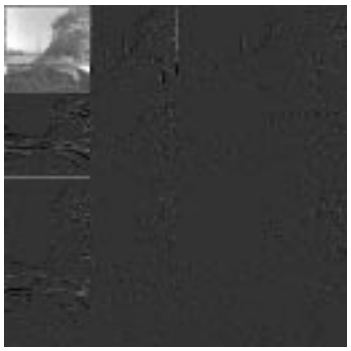

*Fig*11. *DWT coefficients*

*Fig*12. *extract*

3.2 Detection

For embedded data to be undetectable, it needs to be transparent in both the spatial and the transform domains. Manjunath et al. [2] proposed the method of embedding under the DWT coefficients, where only the coefficients in the middle frequency range are used. That is, in Figure 11, embedding involves all frequency bands except the area of the left upper corner (corresponding to lower frequency bands) and the right lower corner (corresponding to the higher frequency band). The cover and the stego images are shown in Figures 13&14 where both images were taken from a publicly available web site [12]. The two show no visual significant difference. At least, they both look legitimate. How-

ever, visual transparency in the spatial domain does not imply UN-detectability. In fact, we can effectively show that embedded information exists in Figure 14. Our detection method is based on frequency domain analysis. We applied the DFT to both the cover and the stego images of Figure 13&14 (only on the Red color byte). Their DFT matrices are shown in Figure 15&16, where to highlight the contrast, only the most significant bit is used in the display. The image with embedded data shows a striking bright diamond pattern that surrounds the center, while the cover image (Figure 13) with no embedding has a common radial shape. Recall that in the DFT display frequency components that correspond to the highest frequency are located in the corner areas, those corresponding to the lowest frequency are in the center, and coefficients on the band of the diamond belong to the middle frequency range. The diamond pattern is also seen in several stego images we have tested.[12] As a result, this seemingly transparent embedding method fails our simple detection test. The embedding technique proposed in [2] is valuable if the stego image of Figure 14 is for watermarking, but not steganography. Note, watermarking *was* the intention of [2].

*Fig*13. *cover*

*Fig*14. *stego*

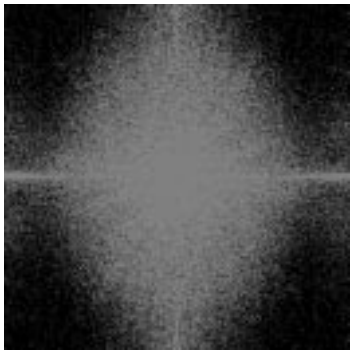

*Fig*15. *DFT of cover*

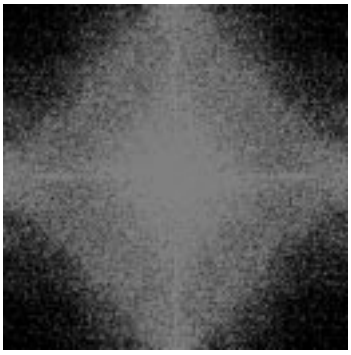

*Fig*16. *DFT of stego*

[12]The diamond shape in some images are not so clear.

So far the strongest result on detection was perhaps made by J. Fridrich at the IHW2001, where she claimed that her method can potentially detect messages as short as any single bit change in a JPEG image.[13] Her method examines whether or not a 8x8 block of JPEG pixels could have been produced by any block of quantized DCT coefficients (also in [6]). This result is interesting because JPEG is frequently used. We are currently analyzing their approach and studying its applicability to other transform methods.

3.3 Robustness

To improve robustness, it may be necessary to reduce the size of embedded data and embed it multiple times under different parts of selected coefficients, where each embedding responds to a particular attack in a different way. Interesting work in robustness was recently reported by [11] (called *cocktail*) and it is one of few methods claimed to be very robust against variety of attacks. The basic observation in [11] is that most attacks will cause magnitudes of more than 50% of frequency coefficients to either increase or decrease. Thus, it makes sense to embed the data twice with one embedding handling the increase and the other embedding handling the decrease. As a result, one embedding is expected to survive with higher chances against any attack.

Can the cocktail embedding method be applied to improve our present NRL L2SB[14] embedding? Unfortunately, it cannot. Recall that NRL L2SB embeds a piece of datum under the least 2 significant bits (so its dynamic range is from 0 to 3) of a pixel whose position is specified by a pre-assigned stego key. The stego key is basically a long-term key and independent of cover images. The cocktail method was designed for watermarking, while NRL L2SB was used to demonstrate the concept of steganography. NRL L2SB is used to extract the embedded message, not just to verify its existence as many watermarking methods do.

We have not yet found a sound method that ensures the robustness of NRL L2SB. In the following, we show a simplistic schemes that may be useful to protect the embedded data against a 2x2 low-pass averaging filtering (e.g., $\begin{bmatrix} 1/4 & 1/4 \\ 1/4 & 1/4 \end{bmatrix}$) and a 2x2 high-pass difference filtering (e.g., $\begin{bmatrix} 1/4 & -1/4 \\ -1/4 & 1/4 \end{bmatrix}$) attacks. Assume position (i,j) of the cover image I is chosen for embedding. Consider the following two cases.

Case 1: average filtering. In the case of averaging filtering, we also embed

[13] The JPEG image generation involves the following steps. For a given input image (I),

- divide the I into a number of 8x8 blocks,
- compute the DCT of each block to yield the DCT coefficient matrix,
- quantize the DCT coefficients,
- evaluate the inverse DCT of the quantized coefficient matrix, and
- round the values to obtain the final JPEG image.

the same datum under the three neighbors (i-1,j-1), (i-1,j) and (i,j-1). The 2x2 averaging filtering computes $\frac{I_s(i-1,j-1)+I_s(i-1,j)+I_s(i,j-1)+I_s(i,j)}{4}$ and stores the result back to position (i,j), where $I_s(.,.)$ denotes the pixel value of the stego at (.,.). As a result, the embedded value at position (i,j) is preserved under this scheme. Note that any pixel value $I_s(k,l)$ in the [0,255] range can be represented as the summation of a multiple of 4 and a remainder, i.e.,

$$I_s(k,l) \;=\; 4m + r \;,$$

where m is a value in [0,63] and r is in [0,3]. If the position (k,l) is selected for embedding, then r denotes the value of the embedded datum. Since each of the four neighbor pixels has the same r, the result of averaging the four pixel values will still have the form, $4m' + r$, with the same r and some number $m' \in [0, 63]$.

Case 2: difference filtering. The difference filtering, which calculates

$$\frac{I_s(i-1,j-1) + I_s(i,j) - I_s(i-1,j) - I_s(i,j-1)}{4},$$

is more involved. In order to preserve the embedded value, we store an embedded value under not one, but two positions. Suppose the embedded value is a "2", which occupies the last two significant bits as 1 and 0 in order from the higher bit to the lower bit. We embed the 1 and the 0 in separate positions.

For "1" embedding, we embed the value 1 under the pixel at (i,j), 0 at (i-1,j), 0 at (i,j-1) and 3 under (i-1,j-1).

For "0" embedding, we embed the value 0 at all four neighbor pixels which have no overlapping with those used for "1" embedding.

This scheme will get the "1" (or "0") back at position (i,j). To extract, two consecutive positions are decoded together.

The length of the stego key under this embedding scheme will increase significantly. The length for embedding against the average filtering becomes 4 times its original length and the length for the case of difference filtering becomes 8 times. Total length is 12 times of the original one. We divide the cover image into two parts at the ratio 1:2 with the smaller part for embedding against the average filtering attack and the larger one for the case of difference filtering. The elongated stego key will inevitably increase the detectability of embedded messages. We are investigating more general robust embedding schemes for steganography. Since in steganography the cover image is usually not available for extraction, robust embedding is a more challenging issue to steganography than to watermarking.

4 FUTURE WORK

Part of our future research will be on the issues of robustness and detectability of information hiding. We showed that a watermarked image which is perceptually invisible in the spatial domain may fail our detectability test. Our approach

to detectability is based on the DFT domain analysis. We proposed a method for protecting data embedded under LSBs against two specific forms of filtering. The two methods need to be refined and expanded for more general applications.

ACKNOWLEDGMENTS: The author thanks Ira S. Moskowitz, Cathy Meadows and John McLean for their helpful comments.

References

[1] AhmedNatarajanRao74 "Discrete Cosine Transform". IEEE Transc. on Computers, Vol c-23, pp 90-93

[2] Chae, J. and Manjunath, B. (1998) " A Robust Embedded Data from Wavelet Coefficients". Proc. SPIE: Storage and Retrieval for Image and Video Databases VI, vol. 3312, pp. 308-317, San Jose, CA, Jan, 1998.

[3] Chang, L. & Moskowitz, I. S. (1997) "Critical Analysis of Security in Voice Hiding Techniques", Proc. in Information and Communication Security, 1997, pp 203-216.

[4] Christopoulos, C., Skodras, A. & Ebrahimi, T. "The JPEG2000 Still Image Coding System: An Overview". IEEE CE, Vol. 46, No. 4, pp. 1103-1127, Nov. 2000.

[5] Cox, I., Kilian, J., Leighton, T. & Shamoon, T. (1995) "Secure Spread Spectrum Watermarking for Multimedia". NEC Research Institute, TR 95-10.

[6] Fridrich, J., Goljan, M. & Du, R. (2001) "Steganalysis Based on JPEG Compatibility," Special session on Theoretical and Practical Issues in Digital Watermarking and Data Hiding, SPIE Multimedia Systems and Applications IV, Denver, CO, August 20-24, 2001.

[7] Gruhl, D., Lu, A. & Bender, W. (1996) "Echo Hiding". In Proceedings of Information Hiding Workshop, University of Cambridge, pp. 295-315.

[8] Johnson, N., Duric, Z., & Jajodia, S. (2001) *Information Hiding: Steganography and Watermarking — attacks and Countermeasures*, Kluwer.

[9] Kang, G. (1985) "Narrowband Integrated Voice Data System Based on the 2400b/s LPC". Naval Research Laboratory Report 8942.

[10] Kurak, C. & McHugh, J. "A Cautionary Note on Image Downgrading". Computer Security Applications Conference, 1992, pp. 153-159.

[11] Lu, C., Huang, S., Sze C. & Liao, H. "Cocktail Watermarking for Digital Image Protection" TR-IIS-99-008.

[12] http://vision.ece.ucsb.edu/watermark/ImageTest2.html.

[13] Marvel, L. "Image Steganography for Hidden Communication". Ph.D. Dissertation, Univ. of Delaware, Dept of EE, 1999.

[14] Moskowitz, I. S., Longdon, G. & Chang, L. "A New Paradigm Hidden in Steganography" New Security Paradigm Workshop 2000, pp. 41-50.

[15] Pennebaker, W. & Mitchell, J. "JPEG STILL IMAGE DATA COMPRESSION STANDARD". van Nostrand Reinhold, 1993.

[16] Sayood, K. (2000) *Data Compression*. Morgan Kaufmann Publishers.

[17] Schneier, B. (1996) *Applied Cryptography*. Wiley.

[18] Smith, J. & Comiskey, B. (1996) "Modulation and Information Hiding in Images". Information Hiding, Springer-Verlag Lecture Notes in Computer Science Volume 1174.

[19] Wolfgang, R., Podilchuk, C. & Delp, E. "Perceptual Watermarks for Digital Images and Video". Proc. of The IEEE, July 1999, pp 1108-1126.

[20] Zeng, W. & Liu, B. (1999) "A Statistical Watermark Detection Technique without Using Original Images for Resolving Rightful Ownerships of Digital Images". IEEE Tran. Image Proc. Nov., 99.